4. RFID Tutorial: Quantifying feeder visits

Gabrielle Davidson - School of Biological Sciences, University of East Anglia

2024-08-15

4. Get set up

Open R Studio

Check R packages installed

```
#load the following packages
library(tidyverse)
library(rmarkdown)
library(dplyr)
library(tidyr)
library(lubridate)
library(ggplot2)
library(stringi) #may not need this
library(magrittr) #may not need this
```

Download resources necessary for the workshop

Download the resources from **my github page (https://github.com/DrGLDavidson/RFID-workshop)**

4.1 Import your dataset to your environment from your working directory.

```
#clear the global environment so we don't have any conflicts with the next steps

rm(list = ls(all.names = TRUE))

#choose the appropriate working directory

setwd("F:/RWorkspace/GitHub/RFID-workshop/data/outputFiles")

#call your most recent dataset

df<-read.delim("Masterdf_noRepeats.txt", header=TRUE)
head(df)
```

```
##         date            dateTime Hmsec     ID Event Channel    Dur Clks   Freq
## 1 2022-02-04 2022-02-04 10:01:13   467 c1935    146       0      0   32 126464
## 2 2022-02-04 2022-02-04 11:06:19   516 c1931  15597       0 127943   26      0
## 3 2022-02-04 2022-02-04 11:06:32   770 c1935    147       0      6   39 126208
## 4 2022-02-04 2022-02-04 11:17:36   871 c1935    148       0 105573   51 126464
## 5 2022-02-04 2022-02-04 11:17:40   123 c1935    148       0 121015   27 126208
## 6 2022-02-04 2022-02-04 11:17:48   543 c1931  15598       0      0   39      0
##   Edges Reps Type  TagID_hex feeder dateRinged timeRinged btoRingType   btoID
## 1   264    2    1 01103FC949    F01 09/01/2021      11:03           R AKL0680
## 2   442    1    1 01103F7DB1    F02 03/10/2021      10:30           R AAJ5894
## 3   426    3    1 01103F7DB1    F01 03/10/2021      10:30           R AAJ5894
## 4  1168    5    1 01103F3BED    F01 17/10/2021      10:09           R AAJ5895
## 5   174    1    1 01103F3BED    F01 17/10/2021      10:09           R AAJ5895
## 6   388    3    1 01103FE3B3    F02 07/11/2021      12:15           R AJT8118
##   species pitTYPE age sex wing weight timeSincePreviousVisit
## 1      BT       R   5   M   64   11.7             firstVisit
## 2      GT       R   3   F   73   16.3             firstVisit
## 3      GT       R   3   F   73   16.3             firstVisit
## 4      GT       R   3   M   75   18.5             firstVisit
## 5      GT       R   3   M   75   18.5                      4
## 6      GT       R   3   F   72   18.1             firstVisit
```

```
names(df)
```

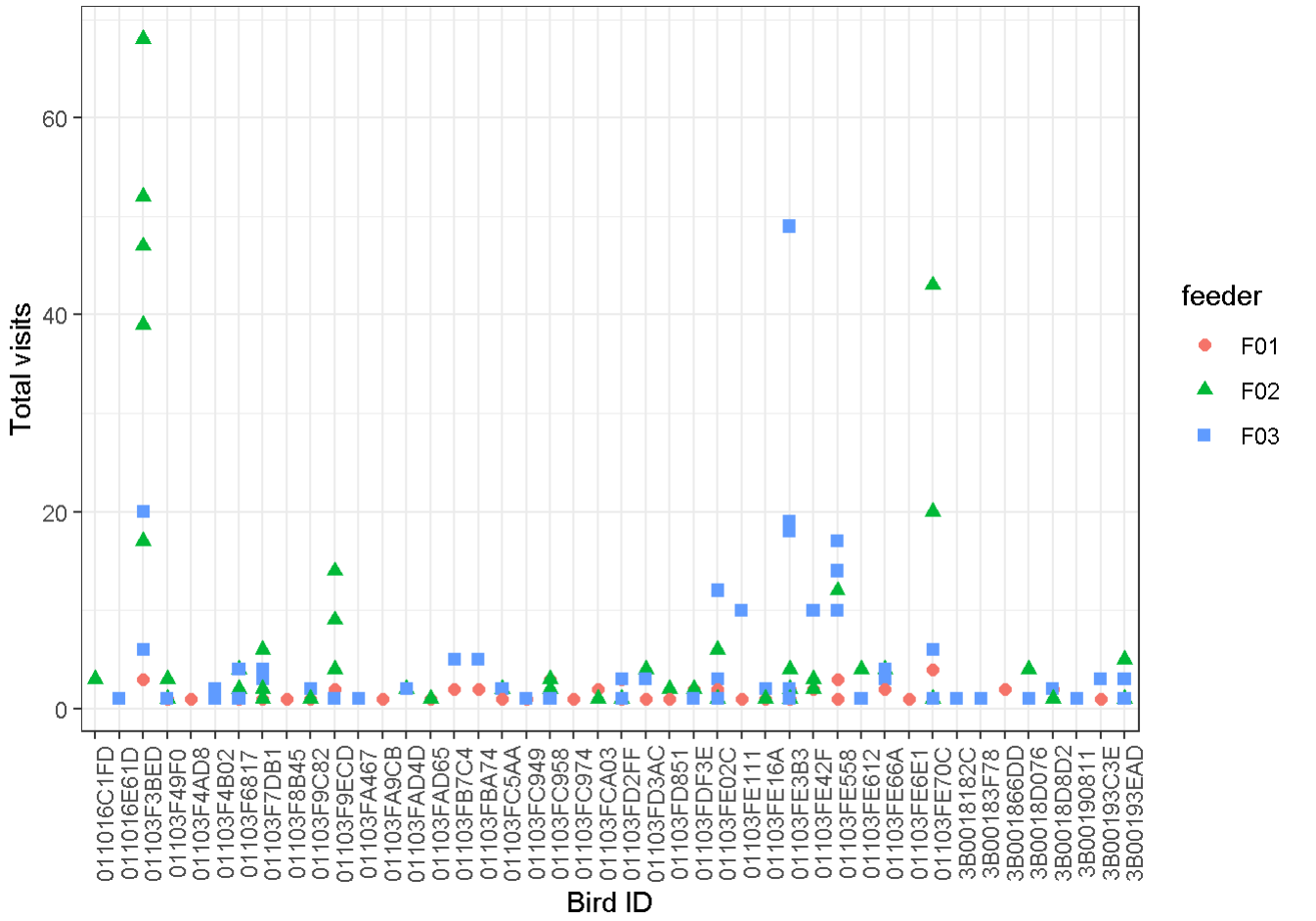
```
##  [1] "date"                  "dateTime"              "Hmsec"
##  [4] "ID"                    "Event"                 "Channel"
##  [7] "Dur"                   "Clks"                  "Freq"
## [10] "Edges"                 "Reps"                  "Type"
## [13] "TagID_hex"             "feeder"                "dateRinged"
## [16] "timeRinged"            "btoRingType"           "btoID"
## [19] "species"               "pitTYPE"               "age"
## [22] "sex"                   "wing"                  "weight"
## [25] "timeSincePreviousVisit"
```

4.2 Total number of visits per feeder, per day

```
#a new dataframe that counts the total number of visits per feeder per individual, per date.
individualVisits<-df%>%
  count(feeder, TagID_hex, date, sort = TRUE)

#rename n column "visits"
names(individualVisits)[names(individualVisits) == "n"] <- "visits"  #rename columns
 names(individualVisits)
```

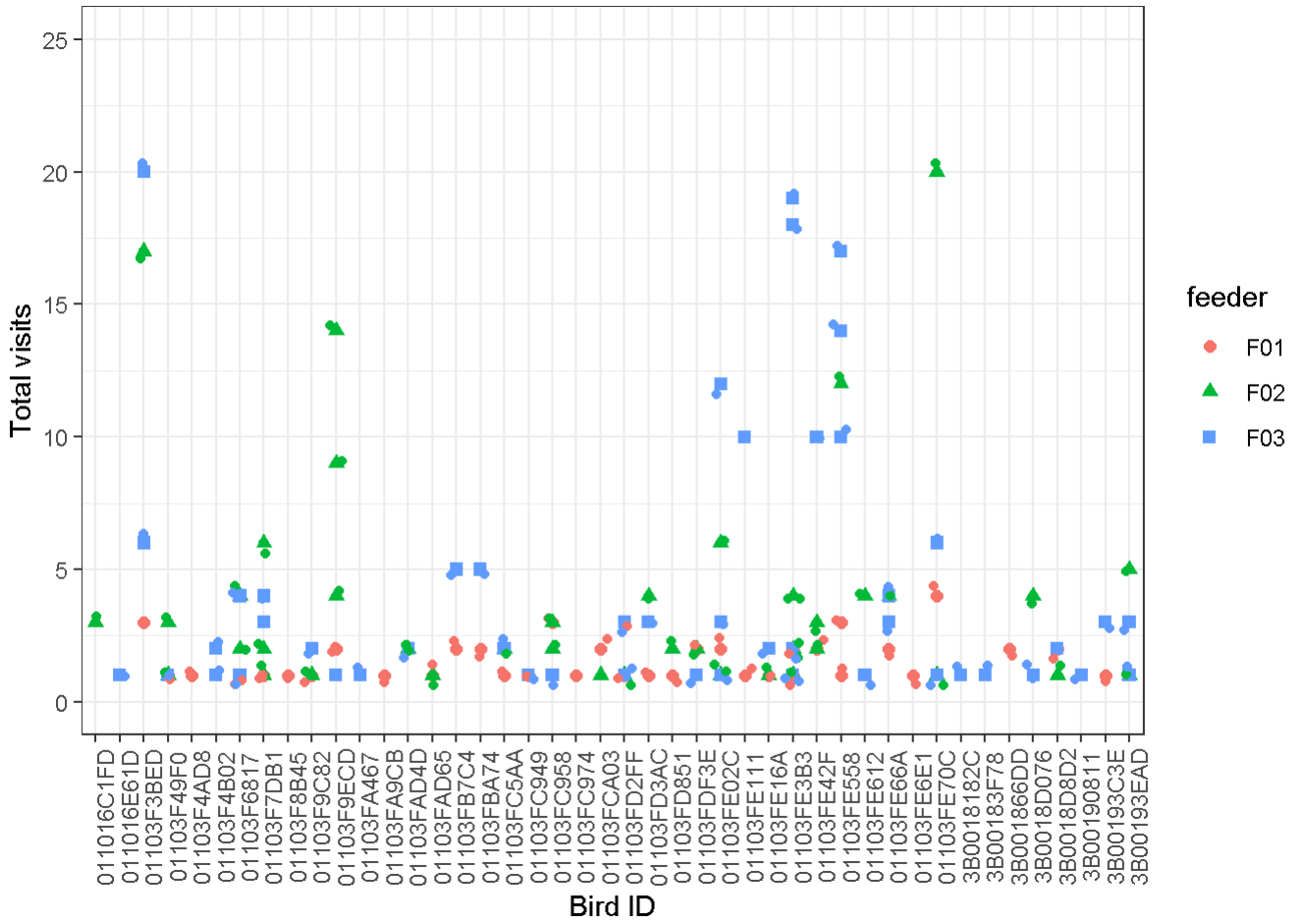
```
## [1] "feeder"    "TagID_hex" "date"      "visits"
```

```
#visualise the data on a figure using ggplot

individualVisits%>%
 ggplot(aes(x = TagID_hex, y = visits, color = feeder))+  #specify axis labels and groupings
  labs(x = "Bird ID", y = "Total visits") + #specify axis labels
  geom_point(aes(shape=feeder), size=2) + # specify shape and size of points for the variable f
eeder
  theme_bw() +   theme(axis.text.x = element_text(angle=90)) #removes the default gray backgrou
nd and rotates x labels.
```



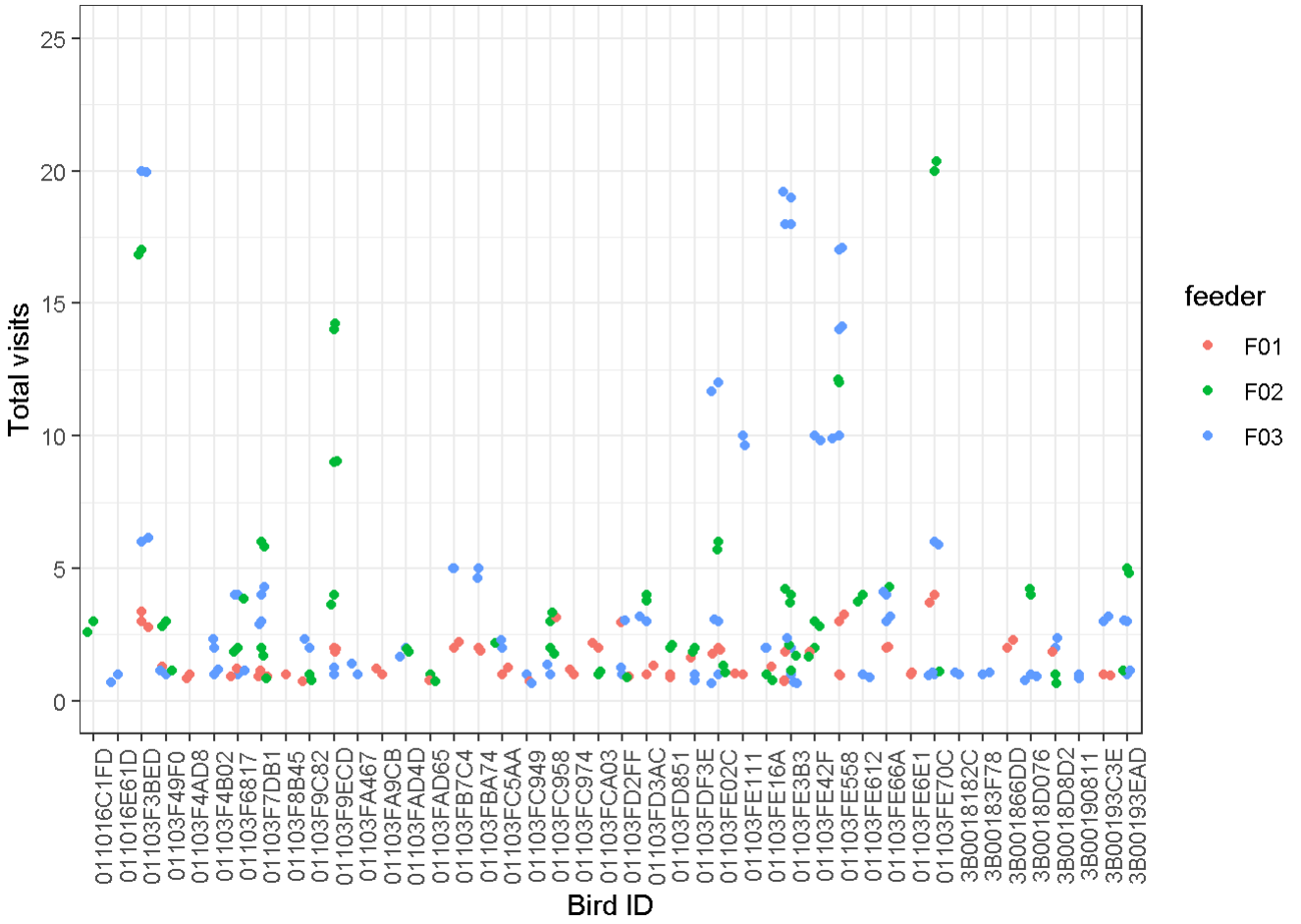

```
#Note that each data point is a unique date. Therefore this plot gives you information not only
on the number of visits they make on a given day, but also over how many days.
#Because of the outliers, it's difficult to resolve what is happening between 0 and 20 visits.

#Restrict axis limits and minimise datapoint overlap

individualVisits%>%
 ggplot(aes(x = TagID_hex, y = visits, color = feeder))+  #specify axis labels and groupings
  labs(x = "Bird ID", y = "Total visits") + #specify axis labels
  geom_point(aes(shape=feeder), size=2) + # specify shape and size of points for the variable f
eeder
  theme_bw() +   theme(axis.text.x = element_text(angle=90))+
  geom_point() + #removes the default gray background and rotates x labels.
 ylim(0, 25) + #specify axis limits
    geom_point(position = position_jitter(width = 0.3)) #jitter the points along a horizontal a
xis to minimise overlap
```

```
## Warning: Removed 7 rows containing missing values (`geom_point()`).
## Removed 7 rows containing missing values (`geom_point()`).
## Removed 7 rows containing missing values (`geom_point()`).
```




```
#can also remove shape for visualization

individualVisits%>%
 ggplot(aes(x = TagID_hex, y = visits, color = feeder))+  #specify axis labels and groupings
  labs(x = "Bird ID", y = "Total visits") + #specify axis labels
  geom_point(size=1) + # specify size ONLY of points for the variable feeder
  theme_bw() +   theme(axis.text.x = element_text(angle=90))+
  geom_point() + #removes the default gray background and rotates x labels.
 ylim(0, 25) + #specify axis limits
    geom_point(position = position_jitter(width = 0.3))
```

```
## Warning: Removed 7 rows containing missing values (`geom_point()`).
## Removed 7 rows containing missing values (`geom_point()`).
## Removed 7 rows containing missing values (`geom_point()`).
```




```
#a new dataframe that counts the proportion an individual visits a feeder relative to the other feeders, per day.
ProportionVisits<-individualVisits%>%
  group_by(TagID_hex, date)%>%
  arrange(TagID_hex)%>%
  mutate(freq = visits / sum(visits))%>%
  ungroup()
```

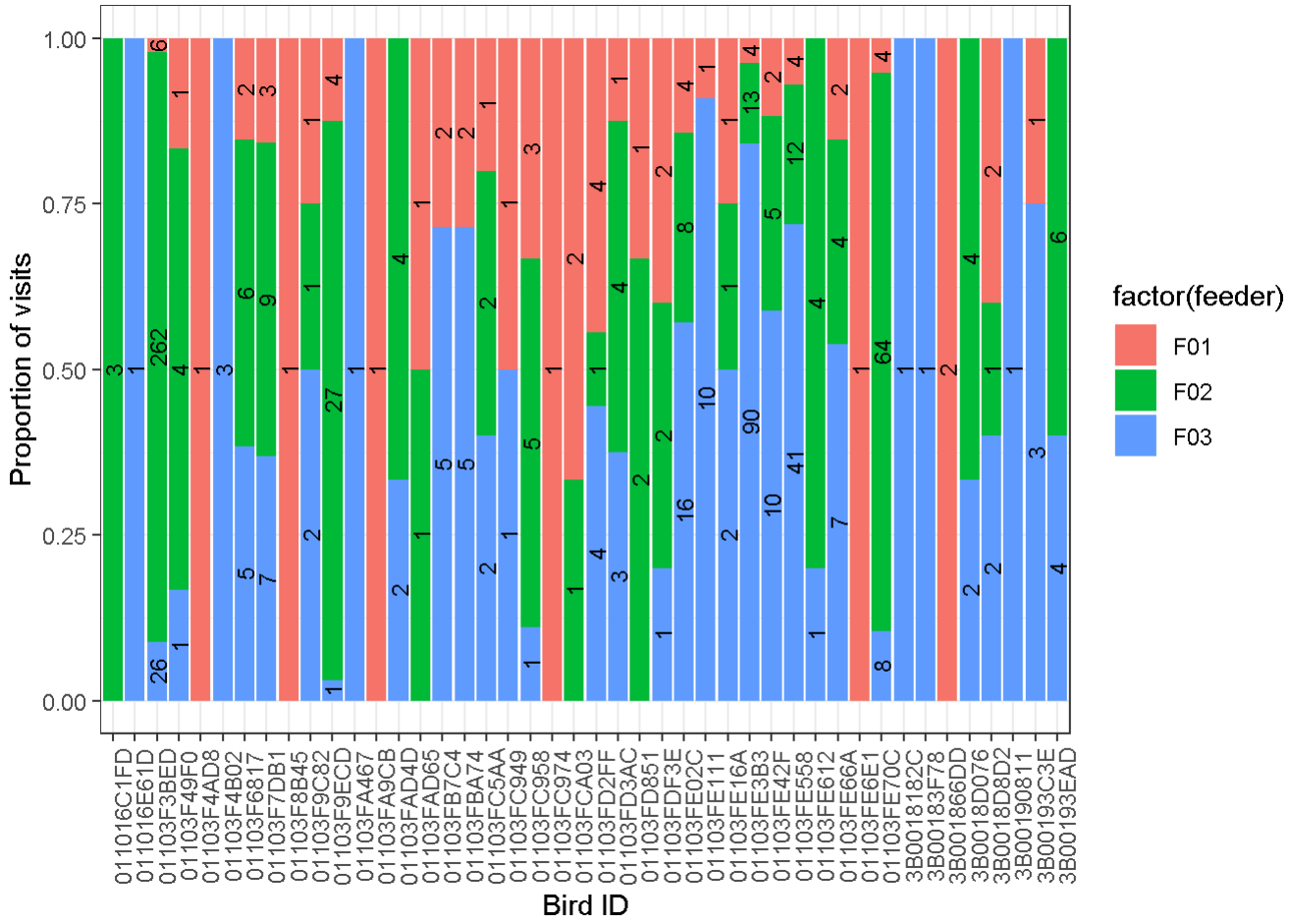
4.3 Proportion of visits per feeder, per day

```
#a new dataframe that counts the proportion an individual visits a feeder relative to the other feeders, per day.
ProportionVisits<-individualVisits%>%
  group_by(TagID_hex, date)%>%
  arrange(TagID_hex)%>%
  mutate(proportion = visits / sum(visits))%>%
  ungroup()


#plot proportions. Better to use the original dataframe as the following code counts each line as a visit

ggplot(df,aes(x=factor(TagID_hex),fill=factor(feeder)))+
  geom_bar(position="fill")+
  geom_text(aes(label=..count..),stat='count',position=position_fill(vjust=0.5), angle=90, size=3) +
  #scale_x_discrete (labels = c("F01","F02","F03")) +
  theme_bw() +   theme(panel.grid.major = element_blank(),
                       panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"),
                       panel.border = element_rect(linetype = "solid", colour = "black", size=.8)) +
  #scale_fill_manual("Feeder", values = c("1" = "gray49", "0" = "gray84"))+
  theme(text=element_text(size=12,  family="serif"),
        axis.ticks = element_line(colour = "black", size = .6)) +
  labs(x = "Bird ID") +
  labs(y = "Proportion of visits") +
  theme_bw() +   theme(axis.text.x = element_text(angle=90))
```

```
## Warning: The `size` argument of `element_rect()` is deprecated as of ggplot2 3.4.0.
## ℹ Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: The `size` argument of `element_line()` is deprecated as of ggplot2 3.4.0.
## ℹ Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## ℹ Please use `after_stat(count)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



4.4 **EXERCISE** Visits

4.4.1 Create a dataframe of the total visits per individual across the whole of your experiment, irrespective of feeder. (hint: group_by)

4.4.2 Create a new dataframe that excludes individuals that visited less than 5 times. (hint: section 3.5)

4.4.3 rename the column of the number of visits to “totalVisits”.

4.4.4 create a figure of your choice representing the number of visits from your new dataframe.

4.4.5 create a dataframe with a list of birds we consider participants and save it as a csv file.

4.5 **EXERCISE** - Time intervals

Previously we created a column that indicated how many seconds had passed since an individual’s previous visit per day. A similar approach can be taken to calculate intervals between visits between individuals (i.e. how long has passed since the previous birds’ visit)

4.5.1 clear your global environment

4.5.2 reload your database “filteredVisitsFeeders.txt”

4.5.3 change the datetime column to be a POSIXct class

4.5.4 create a new column that calculates the time difference from the previous row of a dataframe grouped by date and feeder. Dont forget to use the argument arrange() so the timedate is sequential in your dataframe

4.5.5 create a dataframe consisting of individuals that visited a feeder equal or less than 2 second after the previous visitor.

4.5.6 create a dataframe of a list of unique individuals that have landed on a feeder within 2 second of the previous visitor

4.6 EXERCISE - Correct and incorrect visits

4.6.1 Create a column called “correctChoice” where visits to F01 contain the character “Y” and visits to the other incorrect feeders contain the character “N” (hint: filter, then cbind)

END OF 4. RFID Tutorial: Quantifying feeder visits

```
sessionInfo()
```

```
## R version 4.2.3 (2023-03-15 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United Kingdom.utf8
## [2] LC_CTYPE=English_United Kingdom.utf8
## [3] LC_MONETARY=English_United Kingdom.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United Kingdom.utf8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] magrittr_2.0.3  stringi_1.7.12  rmarkdown_2.22  lubridate_1.9.2
##  [5] forcats_1.0.0   stringr_1.5.0   dplyr_1.1.2     purrr_1.0.1
##  [9] readr_2.1.4     tidyr_1.3.0     tibble_3.2.1    ggplot2_3.4.2
## [13] tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] highr_0.10       bslib_0.5.0      compiler_4.2.3   pillar_1.9.0
##  [5] jquerylib_0.1.4  tools_4.2.3      digest_0.6.31    timechange_0.2.0
##  [9] jsonlite_1.8.4   evaluate_0.21    lifecycle_1.0.3  gtable_0.3.3
## [13] pkgconfig_2.0.3  rlang_1.1.0      cli_3.6.1        rstudioapi_0.14
## [17] yaml_2.3.7       xfun_0.39        fastmap_1.1.1    withr_2.5.0
## [21] knitr_1.43       hms_1.1.3        generics_0.1.3   sass_0.4.6
## [25] vctrs_0.6.1      grid_4.2.3       tidyselect_1.2.0 glue_1.6.2
## [29] R6_2.5.1         fansi_1.0.4      farver_2.1.1     tzdb_0.4.0
## [33] scales_1.2.1     htmltools_0.5.5  colorspace_2.1-0 labeling_0.4.2
## [37] utf8_1.2.3       munsell_0.5.0    cachem_1.0.8     crayon_1.5.2
```